

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
1.Abend • Sie können die Methoden <code>__init__</code> <code>__str__</code> in einer Klasse erläutern und deren Einsatzmöglichkeiten aufzeigen. (K2) • Sie können erläutern was eine ENUM ist. (K2)			
Bruchklasse <ul style="list-style-type: none"> • <code>__init__</code>, <code>__str__</code> • properties • private / public / properties • statische Elemente • API docs / Doc-Strings • Enumerations • sub-classes 	Test-Driven Approach mit Reviews und Refactoring Theoretischen Einschü- ben	200'	Bruchrechner ohne Operator-Overloading fertigstellen

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
2.Abend <ul style="list-style-type: none"> • Sie können eine eigene Klasse designen, implementieren und testen. (K3) • Sie können die Sichtbarkeitsstufen von Methoden und Properties richtig nutzen. (K3) • Sie können ENUMS sinnvoll nutzen. (K3) 			
Operator-Overloading <ul style="list-style-type: none"> • * and ** parameters • Refactoring Logger-Class • Bruch-Klasse (Operator Overloading) 	Test-Driven Approach mit Reviews und Refactoring Theoretischen Einschü- ben	200'	Bruchrechner mit Operator-Overloading fertigstellen

SC\Schulungsunterlagen\HBU\04_PYT2\Unterrichtsplan_HFE_PYT2.docx

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>4.Abend</p> <ul style="list-style-type: none"> Sie können komplexere Applikationen in Python schreiben und Konzepte von Sub-Classing, Multithreading, ENUMS und Template-Mechanism (JINJA) verwenden. (K3) Sie können einen http Request und eine JSON/XML-Response beschreiben. (K2) Sie können API-Dokumentationen analysieren und daraus ableiten, wie externe Pakete in eigene Anwendungen integriert werden können. (K4) 			
<p><u>Multithreading</u></p> <ul style="list-style-type: none"> fork and join time-events 	<p>Test-Driven Approach mit Reviews und Refactoring</p> <p>Theoretischen Einschüben</p>	200'	

SC\Schulungsunterlagen\HBU\04_PYT2\Unterrichtsplan_HFE_PYT2.docx

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
6.Abend • Sie können öffentliche REST-Services analysieren und in eigenen Python-Applikationen nutzen. (K4).			
<u>FLASK Web-Apps</u> • Anotations • Request / Response • static HTML / templates • Mime-Type (JSON Responses) • Parameter Uebergabe (get/put) • Endpoints	Test-Driven Approach mit Reviews und Refactoring Theoretischen Einschü- ben	200'	
7.Abend • Sie können die Vor- und Nachteile von service-orientierter gegenüber funktionaler Programmierung in Python bewerten. (K5)			

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<u>FLASK Web-Apps (Cont.)</u> <ul style="list-style-type: none">• Anotations• Request / Response• static HTML / templates• Mime-Type (JSON Responses)• Parameter Uebergabe (get/put)• Endpoints	Test-Driven Approach mit Reviews und Refactoring Theoretischen Einschü- ben	200'	

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
<p>8.Aband</p> <ul style="list-style-type: none"> Sie können die Effizienz und Sicherheit verschiedener Methoden für den Zugriff auf Raspberry Pi-Systeme beurteilen. (K5) 			
<p><u>Leistungsnachweis (Modullernzielkontrolle MILZ):</u></p> <p>Eine Aufgabe unter Zeitdruck gemäss Spezifikationen nach dem Test-Driven Approach implementieren.</p>	<p>Implementation eines REST Services mit FLASK und JINJA.</p>	<p>200'</p>	

Thema / Inhalt	Methode	Zeit- bedarf	Hausaufgaben
9.Aband <ul style="list-style-type: none">Sie können den Einsatz von Sense-HAT-Modulen und PiPlates in einem Projekt evaluieren und deren Nutzen für spezifische Anwendungen bewerten.(K5)			
Fachgespräche über MLZ <ul style="list-style-type: none">GPIO (LED-Steuerung)PiPlatesShellys Home-Automation	Einzelgespräche gemäss Zeitplan Selbstorganisiertes Lernen (SOL)	200‘	

- Bemerkungen:**
- Jeder Abend dauert 4 Lektionen.
 - Der Unterrichtsplan kann bei Bedarf dem vorhandenen Wissen der Klasse angepasst werden.
 - Die Studierenden lösen die Übungen auf ihren privaten Notebooks und dem eigenen RaspberryPi.
 - Der Leistungsnachweis (**MLZ**) am 8.Aband ist in Einzelarbeit in der vorgegebenen Zeit zu erstellen